

## ЛАБОРАТОРНА РОБОТА № 6

### Поведінкові шаблони

**Мета роботи:** навчитися реалізовувати структурні шоблони проєктування  
**Ланцюжок відповідальностей, Посередник, Спостерігач, Стратегія**

**Хід роботи:**

**Завдання 1:**

#### **Завдання 1: Ланцюжок відповідальностей.**

1. Створіть систему підтримки користувачів.
2. Вона має функціонувати так, як покрокове меню (наприклад, коли телефонуєте до оператора мобільного зв'язку). Але замість голосових повідомлень Ви маєте виводити відповідні повідомлення в консоль і чекати на вибір користувача.
3. Система повинна мати мінімум 4 рівні. Всі питання мають на меті обрати правильний рівень підтримки (тобто Handler) для користувача. Тобто вже на першому питанні може бути підібрано правильний рівень підтримки, тоді меню закінчується. А може бути так, що на жодному питанні не буде знайдено правильний рівень (Handler), тоді меню має повторитися.
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

**Лістинг Program.cs:**

```
// Chain of responsibility
Console.WriteLine("\t---Chain of responsibility---");

Handler h1 = new ConcreteHandler1();
Handler h2 = new ConcreteHandler2();
Handler h3 = new ConcreteHandler3();
Handler h4 = new ConcreteHandler4();

h1.SetNext(h2);
h2.SetNext(h3);
h3.SetNext(h4);
h4.SetNext(null);

while (true)
{
    Console.WriteLine("Choose level (1-4)");

    int level;

    if (!int.TryParse(Console.ReadLine(), out level) || level < 1 || level > 4)
    {
        Console.WriteLine("Bad level input");
        continue;
    }
}
```

					ДУ«Житомирська політехніка».24.121.10.000–Лр 6			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Ганчевський О.О			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Фант М. О.						1
Керівник								11
Н. контр.							ФІКТ Гр. ІПЗ-22-2[1]	
Зав. каф.								

```

        h1.HandleRequest(level);
        break;
    }
}

```

### Лістинг Handler.cs:

```

namespace ClassLibrary.Chain_of_responsibility
{
    public abstract class Handler
    {
        protected Handler? nextHandler;

        public void SetNext(Handler nextHandler)
        {
            this.nextHandler = nextHandler;
        }

        public abstract void HandleRequest(int level);
    }
}

```

### Лістинг ConcreteHandler1.cs:

```

namespace ClassLibrary.Chain_of_responsibility
{
    public class ConcreteHandler1 : Handler
    {
        public override void HandleRequest(int level)
        {
            if (level == 1)
                Console.WriteLine("Hi from level 1");
            else if (nextHandler != null)
                nextHandler.HandleRequest(level);
        }
    }
}

```

### Лістинг ConcreteHandler2.cs:

```

namespace ClassLibrary.Chain_of_responsibility
{
    public class ConcreteHandler2 : Handler
    {
        public override void HandleRequest(int level)
        {
            if (level == 2)
                Console.WriteLine("Hi from level 2");
            else if (nextHandler != null)
                nextHandler.HandleRequest(level);
        }
    }
}

```

### Лістинг ConcreteHandler3.cs:

```

namespace ClassLibrary.Chain_of_responsibility
{
    public class ConcreteHandler3 : Handler
    {
        public override void HandleRequest(int level)
        {
            if (level == 3)
                Console.WriteLine("Hi from level 3");
            else if (nextHandler != null)
                nextHandler.HandleRequest(level);
        }
    }
}

```

		Ганчевський О.О.			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}
}
```

### Лістинг ConcreteHandler4.cs:

```
namespace ClassLibrary.Chain_of_responsibility
{
    public class ConcreteHandler4 : Handler
    {
        public override void HandleRequest(int level)
        {
            if (level == 4)
                Console.WriteLine("Hi from level 4");
            else if (nextHandler != null)
                nextHandler.HandleRequest(level);
        }
    }
}
```

```
---Chain of responsibility---
Choose level (1-4)
q
Bad level input
Choose level (1-4)
7
Bad level input
Choose level (1-4)
3
Hi from level 3
```

### Завдання 2:

#### Завдання 2: Посередник.

1. Відрефакторте [код](#) продемонстрований на лекції за допомогою використання шаблону Посередник.
2. В результаті рефакторингу Aircraft не повинен “знати” про Runway і навпаки. Обидві сутності повинні “знати” лише про CommandCentre.
3. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

### Лістинг Program.cs:

```
// Mediator
Console.WriteLine("\t---Mediator---");

var runway1 = new Runway();
var runway2 = new Runway();

var aircraft1 = new Aircraft("Aircraft 1", 1);
var aircraft2 = new Aircraft("Aircraft 2", 2);
var aircraft3 = new Aircraft("Aircraft 3", 3);

var commandCentre = new CommandCentre(new Runway[] { runway1, runway2 }, new Aircraft[] {
    aircraft1, aircraft2, aircraft3 });

commandCentre.RequestLanding(aircraft1);
commandCentre.RequestLanding(aircraft2);
commandCentre.RequestLanding(aircraft3);

commandCentre.RequestTakeOff(aircraft1);
commandCentre.RequestTakeOff(aircraft1);
```

		Ганчевський О.О			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

### Лістинг Aircraft.cs:

```
namespace ClassLibrary.Mediator
{
    public class Aircraft
    {
        public string Name;
        public bool IsTakingOff { get; set; } = false;
        public Aircraft(string name, int size) => this.Name = name;
        public void Land() {
            this.IsTakingOff = true;

            Console.WriteLine($"Aircraft {this.Name} is landing."); }
        public void TakeOff() {
            this.IsTakingOff = false;
            Console.WriteLine($"Aircraft {this.Name} is taking off."); }
    }
}
```

### Лістинг Runway.cs:

```
namespace ClassLibrary.Mediator
{
    public class Runway
    {
        public readonly Guid Id = Guid.NewGuid();
        public bool IsAvailable { get; set; } = true;
    }
}
```

### Лістинг CommandCentre.cs:

```
namespace ClassLibrary.Mediator
{
    public class CommandCentre
    {
        private List<Runway> _runways = new List<Runway>();
        private List<Aircraft> _aircrafts = new List<Aircraft>();

        public CommandCentre(Runway[] runways, Aircraft[] aircrafts)
        {
            _runways.AddRange(runways);
            _aircrafts.AddRange(aircrafts);
        }

        public void RequestLanding(Aircraft aircraft)
        {
            Console.WriteLine($"Aircraft {aircraft.Name} is requesting landing");

            foreach (var runway in _runways)
            {
                if(aircraft.IsTakingOff)
                {
                    Console.WriteLine($"Aircraft {aircraft.Name} is taking off, cannot land.");
                    return;
                }
                if (runway.IsAvailable)
                {
                    runway.IsAvailable = false;
                    aircraft.Land();
                    Console.WriteLine($"Aircraft {aircraft.Name} has landed on runway {runway.Id}");
                    return;
                }
            }
        }
    }
}
```

		Ганчевський О.О			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

        Console.WriteLine($"Could not land, all runways are busy.");
    }

    public void RequestTakeOff(Aircraft aircraft)
    {
        Console.WriteLine($"Aircraft {aircraft.Name} is requesting takeoff");

        if(!aircraft.IsTakingOff)
        {
            Console.WriteLine($"Aircraft {aircraft.Name} is not taking off, cannot
take off.");
            return;
        }

        foreach (var runway in _runways)
        {
            if (!runway.IsAvailable)
            {
                runway.IsAvailable = false;
                aircraft.TakeOff();
                Console.WriteLine($"Aircraft {aircraft.Name} has taken off from
runway {runway.Id}");
                return;
            }
        }

        Console.WriteLine($"Could not take off, all runways are busy.");
    }
}

```

```

---Mediator---
Aircraft Aircraft 1 is requesting landing
Aircraft Aircraft 1 is landing.
Aircraft Aircraft 1 has landed on runway 6162cf9e-8019-4072-9fc7-fe78048715b8
Aircraft Aircraft 2 is requesting landing
Aircraft Aircraft 2 is landing.
Aircraft Aircraft 2 has landed on runway 32ae44f0-c405-4db8-8e3f-591559a958a6
Aircraft Aircraft 3 is requesting landing
Could not land, all runways are busy.
Aircraft Aircraft 1 is requesting takeoff
Aircraft Aircraft 1 is taking off.
Aircraft Aircraft 1 has taken off from runway 6162cf9e-8019-4072-9fc7-fe78048715b8
Aircraft Aircraft 1 is requesting takeoff
Aircraft Aircraft 1 is not taking off, cannot take off.

```

### Завдання 3:

#### Завдання 3: Спостерігач.

1. Додайте до гри [Blackjack](#) можливість слідкувати за певними подіями в грі.
2. Створіть сервіс, який буде моніторити випадки перебору очків (більше 21) і писати відповідні випадки в окремий файл.
3. Створіть сервіс, який буде збирати аналітику про середньо набрану кількість очок і писати в окремий файл.
4. Для виконання цього завдання створіть PR (він має бути без конфліктів) і відправте посилання на нього.

		Ганчевський О.О.			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

Скриншот програми, та лістинги Program.cs та Game.cs будуть надані у  
ніступному завданні

#### Лістинг IObservable.cs:

```
namespace ClassLibrary.Blackjack.Observer
{
    public interface IObservable
    {
        void AddObserver(IObserver observer);
        void RemoveObserver(IObserver observer);
        void NotifyObservers(string message);
    }
}
```

#### Лістинг IObserver.cs:

```
namespace ClassLibrary.Blackjack.Observer
{
    public interface IObserver
    {
        void Update(string message);
    }
}
```

#### Лістинг EventLogger.cs:

```
namespace ClassLibrary.Blackjack.Observer
{
    public class EventLogger : IObserver
    {
        private string filePath;

        public EventLogger(string filePath)
        {
            this.filePath = filePath;
        }

        public void Update(string message)
        {
            using (var writer = new StreamWriter(filePath, true))
            {
                writer.WriteLine(message);
            }
        }
    }
}
```

#### Лістинг GameObservable.cs:

```
namespace ClassLibrary.Blackjack.Observer
{
    public class GameObservable : IObservable
    {
        private List<IObserver> observers = new List<IObserver>();

        public void AddObserver(IObserver observer)
        {
            this.observers.Add(observer);
        }

        public void RemoveObserver(IObserver observer)
        {
            this.observers.Remove(observer);
        }
    }
}
```

		Ганчевський О.О.			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    public void NotifyObservers(string message)
    {
        foreach (var observer in this.observers)
        {
            observer.Update(message);
        }
    }
}

```

### Лістинг PointsStatistics.cs:

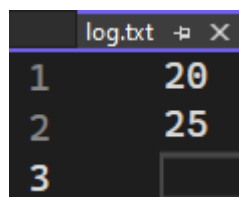
```

namespace ClassLibrary.Blackjack.Observer
{
    public class PointsStatistics : IObserver
    {
        private string filePath;
        private int totalPointsCount;
        private int gamesPlayedCount;

        public PointsStatistics(string filePath)
        {
            this.filePath = filePath;
            totalPointsCount = 0;
            gamesPlayedCount = 0;
        }

        public void Update(string message)
        {
            totalPointsCount += int.Parse(message);
            gamesPlayedCount++;
            using (var writer = new StreamWriter(filePath))
            {
                writer.WriteLine($"Average points: {totalPointsCount /
(double)gamesPlayedCount}");
            }
        }
    }
}

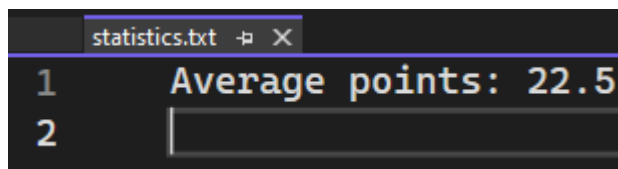
```



```

log.txt
1      20
2      25
3      

```



```

statistics.txt
1      Average points: 22.5
2      

```

### Завдання 4:

		Ганчевський О.О.			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		



#### Завдання 4: Стратегія.

1. Додайте до гри [Blackjack](#) комп'ютерного персонажа.
2. За допомогою патерну Стратегія реалізуйте динамічний вибір опонента.
3. За допомогою патерну Стратегія реалізуйте різні алгоритми гри комп'ютерного персонажу: обережний (буде зупинятися після 13 набраних очок), ризиковий (після 19), рандомний (за Вашим власним алгоритмом).
4. Для виконання цього завдання створіть PR (він має бути без конфліктів) і відправте посилання на нього.

#### Лістинг Program.cs:

```
// Observer/Strategy
Console.WriteLine("\t---Observer/Strategy---");

Game MyGame = new Game();
MyGame.Start();
```

#### Лістинг IStrategy.cs:

```
namespace ClassLibrary.Blackjack.Strategy
{
    public interface IStrategy
    {
        void Play(Player player, GameState state);
    }
}
```

#### Лістинг CautiousStrategy.cs:

```
namespace ClassLibrary.Blackjack.Strategy
{
    public class CautiousStrategy : IStrategy
    {
        public void Play(Player player, GameState state)
        {
            int playerPoints = PointsCounter.CountSum(player.DrawnCards);
            while (playerPoints < 13 && player.ConfirmNextDraw())
            {
                player.DrawCard(state.Deck);
                playerPoints = PointsCounter.CountSum(player.DrawnCards);
            }
        }
    }
}
```

#### Лістинг RandomStrategy.cs:

```
namespace ClassLibrary.Blackjack.Strategy
{
    public class RandomStrategy : IStrategy
    {
        public void Play(Player player, GameState state)
        {
            Random random = new Random();
            int playerPoints = PointsCounter.CountSum(player.DrawnCards);
```

		Ганчевський О.О.			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        while (playerPoints < 21 && player.ConfirmNextDraw())
        {
            if (playerPoints > 17 && random.NextDouble() < 0.5)
                break;

            player.DrawCard(state.Deck);
            playerPoints = PointsCounter.CountSum(player.DrawnCards);
        }
    }
}

```

### Лістинг RiskyStrategy.cs:

```

namespace ClassLibrary.Blackjack.Strategy
{
    public class RiskyStrategy : IStrategy
    {
        public void Play(Player player, GameState state)
        {
            int playerPoints = PointsCounter.CountSum(player.DrawnCards);
            while (playerPoints < 19 && player.ConfirmNextDraw())
            {
                player.DrawCard(state.Deck);
                playerPoints = PointsCounter.CountSum(player.DrawnCards);
            }
        }
    }
}

```

### Лістинг Player.cs:

```

namespace ClassLibrary.Blackjack
{
    public class Player
    {
        public string Name { get; }
        public List<Card> DrawnCards { get; } = new List<Card>();
        public IStrategy Strategy { get; set; }
        public Player(string name, IStrategy strategy)
        {
            this.Name = name;
            Strategy = strategy;
        }
        public bool ConfirmNextDraw()
        {
            return InputHandler.Confirm("Do you want to draw next card?");
        }
        public Card DrawCard(CardsDeck cardsDeck)
        {
            Card card = cardsDeck.Draw();
            this.DrawnCards.Add(card);
            Logger.ShowDrawnCard(card, PointsCounter.CountSum(this.DrawnCards));
            return card;
        }
        public void Play(GameState state)
        {
            this.Strategy.Play(this, state);
        }
    }
}

```

		Ганчевський О.О			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

## Лістинг Game.cs:

```
namespace ClassLibrary.Blackjack
{
    public class Game : GameObservable
    {
        public static readonly int PLAYER_COUNT = 2;
        public static readonly int CARDS_WITHOUT_CONFIRMATION_COUNT = 2;
        private GameState _state = new GameState();

        public Game()
        {
            AddObserver(new EventLogger("log.txt"));
            AddObserver(new PointsStatistics("statistics.txt"));
        }
        private List<Player> _createPlayers()
        {
            var players = new List<Player>();
            for (int i = 1; i <= PLAYER_COUNT; i++)
            {
                string defaultName = $"Player {i}";
                string name = InputHandler.RequestAnswer($"Write a name for [{defaultName}] ", defaultName);
                Player player = new Player(name, ChooseStrategy());
                players.Add(player);
            }
            return players;
        }
        private void _greet()
        {
            Logger.Greet();
        }
        private void _initiateState()
        {
            this._state.SetPlayers(this._createPlayers());
        }
        public void Start()
        {
            this._greet();
            this._initiateState();
            do
            {
                if (this._state.ActivePlayer == null)
                {
                    throw new Exception("No active player detected!");
                }
                this.HandlePlayer(this._state.ActivePlayer);
            }
            while (this._state.SwitchPlayer());
            this.End();
        }

        public void End()
        {
            List<Player> winners = this._state.GetWinners();
            Logger.EndGame(winners);
        }

        public void HandlePlayer(Player player)
        {
            Logger.StartPlayersTurn(player.Name);
            for (int i = 0; i < CARDS_WITHOUT_CONFIRMATION_COUNT; i++)
            {
                player.DrawCard(this._state.Deck);
            }
        }
    }
}
```

		Ганчевський О.О.			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        player.Play(this._state);
        NotifyObservers(PointsCounter.CountSum(player.DrawnCards).ToString());
    }
    public IStrategy ChooseStrategy()
    {
        int strategyChoice = int.Parse(InputHandler.RequestAnswer("Choose a strategy:
1 - Cautious, 2 - Risky, 3 - Random, any - Random."));
        switch (strategyChoice)
        {
            case 1:
                return new CautiousStrategy();
            case 2:
                return new RiskyStrategy();
            case 3:
                return new RandomStrategy();
            default:
                return new RandomStrategy();
        }
    }
}

```

```

---Observer/Strategy---
=====
Hello, let's play BlackJack
=====

Write a name for [Player 1]
Vladyslav

Choose a strategy: 1 - Cautious, 2 - Risky, 3 - Random, any - Random.
2
Write a name for [Player 2]
Monstr

Choose a strategy: 1 - Cautious, 2 - Risky, 3 - Random, any - Random.
3
Now it's Vladyslav's turn!
Get ready

You have drawn King of Clubs
Your total points count is: 10

You have drawn Queen of Clubs
Your total points count is: 20

Switching to player Monstr...
Now it's Monstr's turn!
Get ready

You have drawn Five of Hearts
Your total points count is: 5

You have drawn Ten of Diamonds
Your total points count is: 15

Do you want to draw next card?
Please type y/N:
y

You have drawn King of Hearts
Your total points count is: 25

Checking winners...
The absolute winner is Vladyslav

```

**Висновки:** я навчився реалізовувати структурні шоблони проєктування  
Ланцюжок відповідальностей, Посередник, Спостерігач, Стратегія

Посилання на репозиторій:

<https://github.com/cah9gan/KPZ>

		Ганчевський О.О			ДУ«Житомирська політехніка».24.121.10.000–Лр 6	Арк.
		Фант М. О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		